ARQUITECTURA Y SISTEMAS OPERATIVOS

Trabajo Práctico N.º 6: Gestión del Sistema Operativo y Procesos

ACTIVIDAD PRÁCTICA: BASH SCRIPTING

Introducción:

Este conjunto de ejercicios está diseñado para que los estudiantes se familiaricen con la escritura de scripts en Bash, el lenguaje de comandos utilizado en sistemas Linux. Cada sección está organizada por tema, con una explicación de los objetivos pedagógicos y una serie de ejercicios con su respectiva solución.

Grupo 1: Creación de scripts y comandos básicos

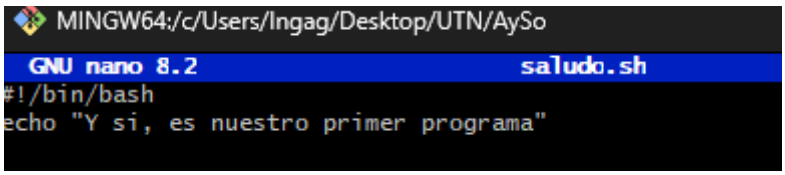
Objetivo: Este primer grupo se enfoca en la introducción a la escritura de scripts en Bash y la ejecución de comandos fundamentales. Los ejercicios buscan que el estudiante se familiarice con la estructura básica de un script, cómo imprimir mensajes en la pantalla, listar el contenido de un directorio y realizar operaciones simples de manipulación de archivos, como crear un directorio y copiar archivos.

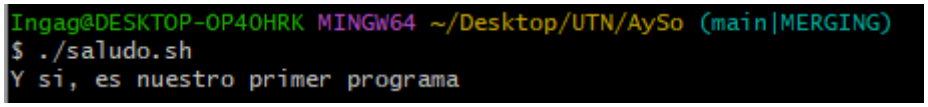
Ejercicio 1.1: Crear un script llamado saludo.sh que muestre en pantalla el mensaje "Y si, es nuestro primer programa".

Ejercicio 1.2: Escribir un script que liste todos los archivos y directorios del directorio actual en formato largo.

Ejercicio 1.3: Crear un script que cree un directorio llamado backup y copie en él todos los archivos .txt del directorio actual.

Ejercicio 1.1:





Ejercicio 1.2:

```
  GNU nano 8.2                          listar.sh
#!/bin/bash
ls -l
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo (main|MERGING)
$ ./listar.sh
total 3704
-rw-r--r-- 1 Ingag 197609 262036 Mar 27 23:25  TP1-AySO-Inga_Gonzalo.docx
-rw-r--r-- 1 Ingag 197609 270771 Mar 27 23:55  TP2-AySO-Inga_Gonzalo.docx
-rw-r--r-- 1 Ingag 197609 166004 Mar 27 23:56  TP2-AySO-Inga_Gonzalo.pdf
-rw-r--r-- 1 Ingag 197609 262786 May  4 16:34  TP3-AySO-Inga_Gonzalo.docx
-rw-r--r-- 1 Ingag 197609 158010 May  4 16:34  TP3-AySO-Inga_Gonzalo.pdf
-rw-r--r-- 1 Ingag 197609 262735 May  4 19:32  TP4-AySO-Inga_Gonzalo.docx
-rw-r--r-- 1 Ingag 197609 158284 May  4 19:32  TP4-AySO-Inga_Gonzalo.pdf
-rw-r--r-- 1 Ingag 197609 831105 May 29 22:22  TP5-AySO-Inga_Gonzalo.docx
-rw-r--r-- 1 Ingag 197609 569319 May 29 22:22  TP5-AySO-Inga_Gonzalo.pdf
-rw-r--r-- 1 Ingag 197609 831105 May 29 22:22  TP6-AySO-Inga_Gonzalo.docx
-rw-r--r-- 1 Ingag 197609      7 May 31 13:50  listar.sh
-rwxr-xr-x 1 Ingag 197609     54 May 31 13:49  saludo.sh
-rw-r--r-- 1 Ingag 197609    162 May 28 19:22 '~$5-AySO-Inga_Gonzalo.docx'
-rw-r--r-- 1 Ingag 197609    162 May 31 13:08 '~$6-AySO-Inga_Gonzalo.docx'
```

Ejercicio 1.3:

```
  GNU nano 8.2                          backup.sh
#!/bin/bash

mkdir -p backup

cp *.txt backup/
```

Grupo 2: Variables y operadores aritméticos

Objetivo: Los ejercicios están diseñados para que el estudiante aprenda a declarar variables, asignarles valores, realizar cálculos matemáticos y mostrar los resultados en la pantalla. Se introduce el uso de la expansión aritmética y, opcionalmente, el uso de bc para cálculos de mayor precisión.

Ejercicio 2.1: Crea un script que defina dos variables numéricas y muestre la suma, resta, multiplicación y división entre ellas.
Ejercicio 2.2: Escribe un script que calcule el área de un rectángulo a partir de dos variables (base y altura).
Ejercicio 2.3: Define tres variables: nombre, edad y ciudad, y luego imprime un mensaje con esos datos concatenados en una oración.

Ejercicio 2.1:

```
  GNU nano 8.2                          operaciones.sh
#!/bin/bash

num1=40
num2=10

suma=$((num1 + num2))
resta=$((num1 - num2))
multiplicacion=$((num1 * num2))
division=$((num1 / num2))

echo "Suma: $suma"
echo "Resta: $resta"
echo "Multiplicación: $multiplicacion"
echo "División: $division"
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo
$ ./operaciones.sh
Suma: 50
Resta: 30
Multiplicación: 400
División: 4
```

Ejercicio 2.2:

```
  GNU nano 8.2                                area.sh
#!/bin/bash

base=10
altura=20

area=$((base*altura))

echo "El area del cuadrado es de: $area"
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo
$ ./area.sh
El area del cuadrado es de: 200
```

Ejercicio 2.3:

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo (main|MERGING)
$ ./datos_personales.sh
Hola, mi nombre es Gonzalo, tengo 26 años y vivo en Buenos Aires.
```

```
  GNU nano 8.2                    datos_personales.sh
#!/bin/bash

# Definir variables
nombre="Gonzalo"
edad=26
ciudad="Buenos Aires"

# Mostrar mensaje concatenado
echo "Hola, mi nombre es $nombre, tengo $edad años y vivo en $ciudad."
```

Grupo 3: Condicionales

Objetivo: Este grupo se centra en la implementación de la lógica condicional en los scripts de Bash, utilizando las estructuras if, then, else y elif. El objetivo es que el estudiante aprenda a escribir scripts que tomen decisiones basadas en diferentes condiciones, como la edad del usuario, la existencia de un archivo o el valor de una nota.

Ejercicio 3.1: Escribir un script que pida al usuario ingresar su edad y muestre si es mayor o menor de edad.

Ejercicio 3.2: Crear un script que verifique si un archivo existe. Si existe, muestra "El archivo existe"; de lo contrario, "El archivo no existe".

Ejercicio 3.3: Realiza un script que evalúe una nota (de 0 a 10) e imprima: "Reprobado" (menos de 6), "Aprobado" (6 a 8), "Excelente" (9 o 10).

Ejercicio 3.1:

```
  GNU nano 8.2                          edad.sh
#!/bin/bash

read -p "Ingresá tu edad: " edad

if [ "$edad" -ge 18 ]; then
    echo "Sos mayor de edad."
else
    echo "Sos menor de edad."
fi
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo
$ ./edad.sh
Ingresá tu edad: 26
Sos mayor de edad.

Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo
$ ./edad.sh
Ingresá tu edad: 16
Sos menor de edad.
```

Ejercicio 3.2:

```
  GNU nano 8.2                                notas.sh
#!/bin/bash

read -p "Ingresá tu nota (0 a 10): " nota

if [ "$nota" -ge 0 ] && [ "$nota" -le 10 ]; then
    if [ "$nota" -lt 6 ]; then
        echo "Reprobado"
    elif [ "$nota" -ge 6 ] && [ "$nota" -le 8 ]; then
        echo "Aprobado"
    else
        echo "Excelente"
    fi
else
    echo "Nota inválida. Debe estar entre 0 y 10."
fi
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo
$ ./notas.sh
Ingresá tu nota (0 a 10): 5
Reprobado

Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo
$ ./notas.sh
Ingresá tu nota (0 a 10): 9
Excelente

Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo
$ ./notas.sh
Ingresá tu nota (0 a 10): 8
Aprobado
```

Grupo 4: Bucles

Objetivo: Se introducen las estructuras de control de bucles: for, while y until. Los ejercicios buscan que el estudiante aprenda a repetir una serie de comandos múltiples veces, ya sea un número fijo de veces o hasta que se cumpla una determinada condición. Se practican la iteración sobre rangos de números, la acumulación de valores y la creación de bucles que se ejecutan hasta que el usuario ingresa una contraseña específica.

Ejercicio 4.1: Crear un script que imprima los números del 1 al 10 usando un bucle for.
Ejercicio 4.2: Escribir un script que sume todos los números del 1 al 100 utilizando un bucle while.
Ejercicio 4.3: Crear un script que pida al usuario ingresar contraseñas hasta que escriba "secreto", usando un bucle until.

Ejercicio 4.1:

```
  GNU nano 8.2                    numeros.sh
#!/bin/bash

for i in {1..10}
do
  echo $i
done
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo
$ ./numeros.sh
1
2
3
4
5
6
7
8
9
10
```

Ejercicio 4.2:

```
  GNU nano 8.2                    sumaNumeros.sh
#!/bin/bash

suma=0
i=1

while [ $i -le 100 ]
do
  suma=$((suma + i))
  i=$((i + 1))
done

echo "La suma es: $suma"
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo
$ ./sumaNumeros.sh
La suma es: 5050
```

Ejercicio 4.3:

```
 GNU nano 8.2                    contrasenia.sh
#!/bin/bash

contrasenia=""

until [ "$contrasenia" = "secreto" ]
do
  read -p "Ingresa la contraseña: " contrasenia
done

echo "¡Contraseña aceptada!"
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo
$ ./contrasenia.sh
Ingresa la contraseña: hola
Ingresa la contraseña: noes
Ingresa la contraseña: secreto
¡Contraseña aceptada!
```

Grupo 5: Entrada del usuario

Objetivo: El foco está en cómo interactuar con el usuario, solicitando y leyendo datos desde la entrada estándar. Los ejercicios cubren la lectura de nombres, palabras y contraseñas, incluyendo el manejo de contraseñas ocultas y la verificación de la coincidencia entre ellas.

Ejercicio 5.1: Crear un script interactivo que solicite nombre y apellido y los imprima en mayúsculas.
Ejercicio 5.2: Escribe un script que pida al usuario una palabra y luego muestre cuántos caracteres tiene.
Ejercicio 5.3: Crear un script que solicite al usuario una contraseña oculta con “read -s” y luego confirme su ingreso con un mensaje

Ejercicio 5.1:

```
 GNU nano 8.2                  nombreMayuscula.sh
#!/bin/bash

read -p "Ingresá tu nombre: " nombre
read -p "Ingresá tu apellido: " apellido


nombreMayus=$(echo "$nombre" | tr '[:lower:]' '[:upper:]')
apellidoMayus=$(echo "$apellido" | tr '[:lower:]' '[:upper:]')

echo "Nombre completo en mayúsculas: $nombreMayus $apellidoMayus"
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo/ejercicios
$ ./nombreMayuscula.sh
Ingresá tu nombre: Gonzalo
Ingresá tu apellido: Inga
Nombre completo en mayúsculas: GONZALO INGA
```

Ejercicio 5.2:

```
  GNU nano 8.2                    contarCaracteres.sh
#!/bin/bash

read -p "Ingresá una palabra: " palabra

longitud=${#palabra}

echo "La palabra '$palabra' tiene $longitud caracteres."
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo/ejercicios
$ ./contarCaracteres.sh
Ingresá una palabra: gonzalo
La palabra 'gonzalo' tiene 7 caracteres.
```

Ejercicio 5.3:

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo/ejercicios
$ ./oculta.sh
Ingresá tu contraseña:
Contraseña ingresada correctamente.
```

```
  GNU nano 8.2                          oculta.sh
#!/bin/bash

read -s -p "Ingresá tu contraseña: " contrasenia
echo
echo "Contraseña ingresada correctamente."
```

Grupo 6: Manipulación de cadenas

Objetivo: Los ejercicios exploran cómo extraer partes de una cadena, reemplazar subcadenas y convertir el texto entre mayúsculas y minúsculas. El objetivo es que el estudiante adquiera habilidades para procesar y modificar texto dentro de los scripts.

Ejercicio 6.1: Escribir un script que tome una cadena y extraiga los primeros 8 caracteres.
Ejercicio 6.2: Crea un script que reemplace todas las ocurrencias de la palabra "error" por "problemita" en una variable.
Ejercicio 6.3: Escribir un script que convierta un texto ingresado por el usuario a minúsculas y lo imprima.

Ejercicio 6.1:

```
  GNU nano 8.2                          contar8.sh
#!/bin/bash

read -p "Ingresá una cadena: " texto

primeros8="${texto:0:8}"

echo "Los primeros 8 caracteres son: $primeros8"
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo/ejercicios
$ ./contar8.sh
Ingresá una cadena: hola como estas
Los primeros 8 caracteres son: hola com
```

Ejercicio 6.2:

```
  GNU nano 8.2                          reemplazar.sh
#!/bin/bash

mensaje="Hubo un error en el sistema. Otro error ocurrió después."

# Reemplazo usando sustitución de Bash
nuevoMensaje=${mensaje//error/problemita}

echo "Original: $mensaje"
echo "Modificado: $nuevoMensaje"
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo/ejercicios (main|MERGING)
$ ./reemplazar.sh
Original: Hubo un error en el sistema. Otro error ocurrió después.
Modificado: Hubo un problemita en el sistema. Otro problemita ocurrió después.
```

Ejercicio 6.3:

```
  GNU nano 8.2                          minusculas.sh
#!/bin/bash

read -p "Ingresá un texto: " texto

minusculas=$(echo "$texto" | tr '[:upper:]' '[:lower:]')

echo "Texto en minúsculas: $minusculas"
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo/ejercicios
$ ./minusculas.sh
Ingresá un texto: MI NOMBRE ES GONZALO
Texto en minúsculas: mi nombre es gonzalo
```

Grupo 7: Scripts combinando conceptos

Objetivo: se proponen ejercicios que integran varios de los conceptos aprendidos en los grupos anteriores. Esto permite al estudiante aplicar un enfoque más completo y complejo en la resolución de problemas, combinando la lógica condicional, los bucles, la entrada del usuario y la manipulación de cadenas para crear scripts más útiles y funcionales.
Ejercicio 7.1: Escribir un script que solicite al usuario su nombre y edad, y luego le diga si puede votar (mayor de 16).
Ejercicio 7.2: Crear un script que lea una lista de nombres desde un archivo de texto (nombres.txt) e imprima un saludo personalizado para cada uno.
Ejercicio 7.3: Generar codigo bash para calcular el promedio entre 5 números utilizando el bucle for.

Ejercicio 7.1:

```
  GNU nano 8.2                    votoValido.sh
#!/bin/bash

read -p "Ingresá tu nombre: " nombre
read -p "Ingresá tu edad: " edad

if [ "$edad" -ge 16 ]; then
    echo "$nombre, puede votar."
else
    echo "$nombre, todavia no puede votar."
fi
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo/ejercicios
$ ./votoValido.sh
Ingresá tu nombre: Gonzalo
Ingresá tu edad: 19
Gonzalo, puede votar.

Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo/ejercicios
$ ./votoValido.sh
Ingresá tu nombre: Alexis
Ingresá tu edad: 15
Alexis, todavia no puede votar.
```

Ejercicio 7.2:

```
  GNU nano 8.2                          saludoLista.sh
#!/bin/bash

# Verificamos si el archivo existe
if [ ! -f nombres.txt ]; then
    echo "El archivo 'nombres.txt' no existe."
    exit 1
fi

while read -r nombre; do
    echo "¡Hola, $nombre!"
done < nombres.txt
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo/ejercicios
$ ./saludoLista.sh
!Hola, Gonzalo
!Hola, Alexis
!Hola, Lautaro
!Hola, Axel
!Hola, Facundo
!Hola, Aixa
!Hola, Fiorella
```

Ejercicio 7.2:

```
  GNU nano 8.2                            promedio.sh
#!/bin/bash

suma=0

for i in {1..5}; do
    read -p "Ingresá el número $i: " num
    suma=$((suma + num))
done

promedio=$((suma / 5))

echo "El promedio es: $promedio"
```

```
Ingag@DESKTOP-OP4OHRK MINGW64 ~/Desktop/UTN/AySo/ejercicios
$ ./promedio.sh
Ingresá el número 1: 20
Ingresá el número 2: 30
Ingresá el número 3: 40
Ingresá el número 4: 60
Ingresá el número 5: 10
El promedio es: 32
```